Abstract

In this note I describe a procedure which allows us to correctly use the current sample of produced crossings to calculate (trigger) rates. The method uses weights to get the best statistical power from the current sample. The method requires some information about each event to correctly compute the weight. Since this information is not yet available, average weights were used to test the correctness and statistical power of the weighting method. The results are promising.

1 Introduction

As discussed in the note “Calculating Trigger Rates in the Presence of Pileup” by S. Eno and P. Sphicas, the problem of computing those trigger rates which may come from the sum of several events in a crossing is difficult. The purpose of this calculation is to make an attempt to use the existing production, or one like it, to compute trigger rates that may be due to the sum of events, with more statistical precision than in the methods proposed in the above mentioned note (which you should read to understand the issues addressed here).

CMS has produced crossings in which the “first event” is constrained to come from some $\hat{P}_T$ bin to which pileup events are added. The pileup events are minimum bias events and have the physically expected distribution of jets including those at high $\hat{P}_T$. There are 17.3 pileup events from the same crossing plus more from other crossings. The “first event” bin distribution produced emphasizes high $\hat{P}_T$ bins which are more likely to be accepted by the triggers. That is, we produced more events in high $\hat{P}_T$ bins than expected from the cross sections so that we would have good statistics on rare processes that trigger us. The goal is to do a calculation with small statistical errors yet avoid double counting due to high $\hat{P}_T$ events in pileup.

To get small statistical errors with the limited sample we have, we cannot rely on the pileup to simulate rare events. There are too many crossings per second. We must rely on the events binned in $\hat{P}_T$. This means we cannot treat the binned events as if they are only the first event; we must use them as if they could be any of the 17.3 events in a crossing.

The approach attempted here is to compute a weight for each event that will allow us to do this.

2 The Calculation of the Weight

Assume we have N_{bin} bins in $\hat{P}_T$ and that a given crossing has

$$(n_1, n_2, ..., n_{N_{bin}}) \equiv \vec{n}$$

events from those bins. The probability to have this distribution of events in bins is

$$P(\vec{n}) = \prod_{i=1}^{N_{bin}} \frac{e^{-f_i\mu}(f_i\mu)^{n_i}}{n_i!}.$$

In the production, we generated N_j crossings in which the first event was required to be in the j^{th} $\hat{P}_T$ bin. Given these numbers N_j, the probability that one of the generated crossings have the distribution in bins given by $\vec{n}$ is

$$\hat{P}(\vec{n}) = \sum_{j=1;n_j\neq 0}^{N_{bin}} \frac{N_j}{N} \frac{e^{-f_j\mu}(f_j\mu)^{n_j-1}}{(n_j-1)!} \prod_{i\neq j} \frac{e^{-f_i\mu}(f_i\mu)^{n_i}}{n_i!}.$$

$$\hat{P}(\vec{n}) = \sum_{j=1;n_j\neq 0}^{N_{bin}} \frac{N_j}{N} \frac{n_j}{f_j\mu} \prod_{i=1}^{N_{bin}} \frac{e^{-f_i\mu}(f_i\mu)^{n_i}}{n_i!} = \sum_{j=1}^{N_{bin}} \frac{N_j}{N} \frac{n_j}{f_j\mu} \prod_{i=1}^{N_{bin}} \frac{e^{-f_i\mu}(f_i\mu)^{n_i}}{n_i!}$$

To weight these events so the they give a trigger rate we need to multiply by the bunch crossing rate, divide by the total events generated, and correct for the relative probabilities.

$$W(\vec{n}) = \frac{32\times 10^6}{N} \frac{P(\vec{n})}{\hat{P}(\vec{n})}$$

where $N = \sum_{j=1}^{N_{bin}} N_j$.

A little arithmetic gives

$$W(\vec{n}) = \frac{32\times 10^6 \mu}{\sum_{j=1}^{N_{bin}} N_j \frac{n_j}{f_j}}.$$

This is the weight to use for each event given $\vec{n}$. Once the $\vec{n}$ is available for production events, we can use this to get a correct and statistically more accurate prediction of the trigger rates.

These weights should be useful for the computation of any trigger rate. We do not need to assume that the trigger is based on one event. The statistical power will be greatly improved compared to the "first event method" of section 6.1 of the Eno and Sphicas note. The weighting method makes no approximation. At this time we do not have the information saved per event to compute the weight. One would hope that we will have this information in the next production, or even better, that it could be recovered for the existing production. (If we have the information, we could also use the weight to remove fluctuations by counting the actual number of events of each type and weighting to the expected.)

3 Average Weights for a Bin

Since we may not have enough information for now, we can try to compute an average weight for each bin that will correct as well as we can for double counting. This is an approximation since all the events in a $\hat{P}_T$ bin will be given the same weight. For processes in which several high $\hat{P}_T$ events conspire to give a trigger, this average weight could lead to an overestimate of the trigger rate. We need to test whether this is true. In any case, the calculation of the average will allow us to test the validity of the weight calculation and will help us determine the statistical power of the method.

For the k^{th} bin,

$$\bar{W}_k = \sum_{\vec{n}} W(\vec{n}) P_k(\vec{n})$$

where $P_k(\vec{n})$ is the probability to get the bin distribution given by $\vec{n}$ for the event we generated in bin k.

$$P_k(\vec{n}) = \frac{e^{-f_k\mu}(f_k\mu)^{n_k-1}}{(n_k-1)!} \prod_{i\neq k} \frac{e^{-f_i\mu}(f_i\mu)^{n_i}}{n_i!}.$$

Plugging this in we get

$$\bar{W}_k = \sum_{\vec{n};n_k\neq 0} \frac{32\times 10^6 \mu}{\sum_{j=1}^{N_{bin}} N_j \frac{n_j}{f_j}} \frac{e^{-f_k\mu}(f_k\mu)^{n_k-1}}{(n_k-1)!} \prod_{i\neq k} \frac{e^{-f_i\mu}(f_i\mu)^{n_i}}{n_i!}$$

$$\bar{W}_k = 32\times 10^6 \mu \sum_{\vec{n};n_k\neq 0} \frac{1}{\sum_{j=1}^{N_{bin}} N_j \frac{n_j}{f_j}} \frac{n_k}{f_k\mu} \prod_{i}^{N_{bin}} \frac{e^{-f_i\mu}(f_i\mu)^{n_i}}{n_i!}$$

$$\bar{W}_k = 32\times 10^6 \sum_{\vec{n}} \frac{n_k}{f_k \sum_{j=1}^{N_{bin}} N_j \frac{n_j}{f_j}} \prod_{i}^{N_{bin}} \frac{e^{-f_i\mu}(f_i\mu)^{n_i}}{n_i!}$$

This can be computed numerically and saved for weighting the distributions. The sum over $\vec{n}$ represents N_{bin} sums from 0 to infinity, but, in practice, only a few terms contribute and the whole calculation can be done an hour on one slow computer.

The table shows the weight for each bin, which has units of Hz. Simply use this weight for each crossing in the bin to get a rate. The "Rate" column

$\hat{P}_T$ Bin	Weight	Rate Hz	Wt. relative to "1st event method"
0-10	112.164	10454229	0.3963
10-15	94.0227	8912189	2.1174
15-20	50.8531	6545203	7.1572
20-30	22.0589	4547049	11.7703
30-50	13.0147	1297412	13.6617
50-80	6.3129	202261	15.3346
80-120	1.9336	31205	16.6349
120-170	0.4768	5492	17.1263
170-230	0.8715	1115	16.9890
230-300	0.2074	266	17.2223
300-380	0.0544	70	17.2780
380-470	0.0155	21	17.2924
Total		31.997	

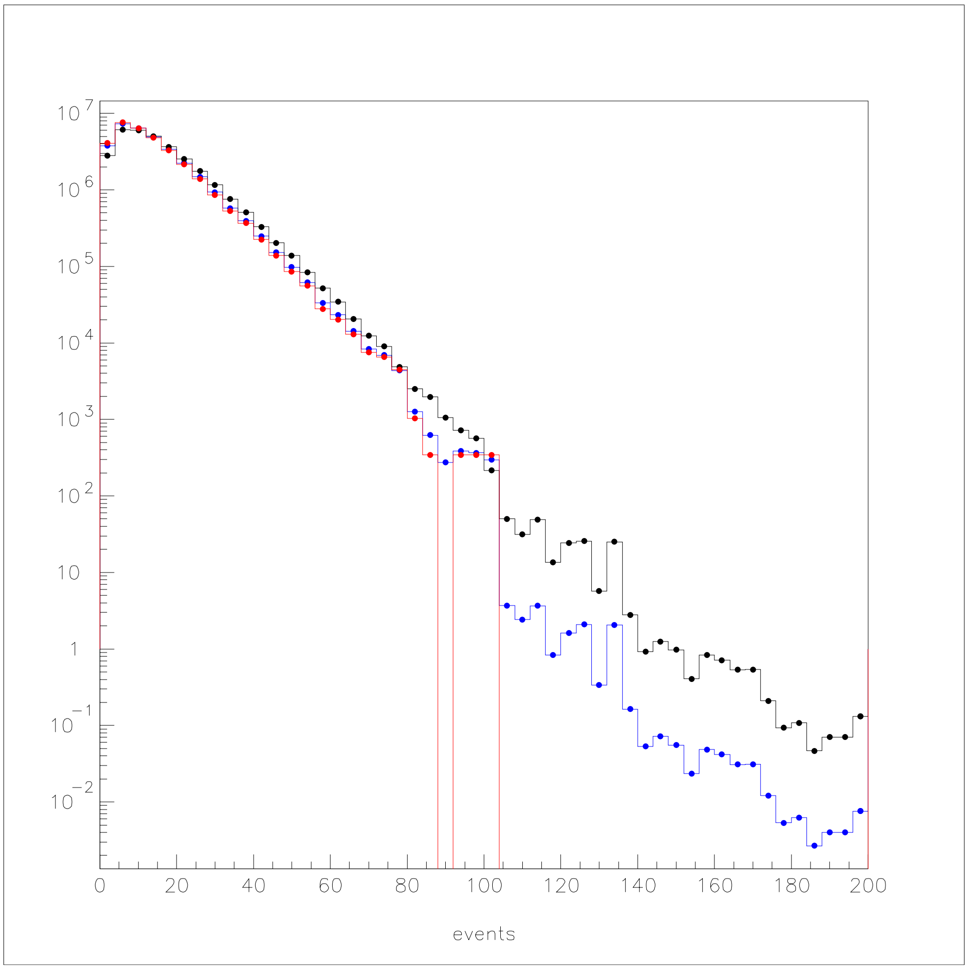
Table 1: Result of average weight calculation

shows the contribution to the total rate (no trigger requirement). It is a check of the method that it adds up to 32 MHz. This clearly indicates that the calculation is working. The last column show the relative weight compared to the "first event method". The relative weight approaches 17.3 in the high bins, which is what I would expect if we can use the high $\hat{P}_T$ event as any one of the 17.3 events. The relative weight is low for the lowest bin indicating we rely less on the pileup.

Since there can only be one set of completely correct weights and Eno and Sphicas have proven the "first event" weights are correct, these weights must not be fully correct. Again this is due to the averaging procedure. They clearly give the right answer for no trigger cut, should give the right answer in the case that the trigger depends on one event from the crossing, but might overestimate the rate in between. Again, if we did not have to average and could apply a weight on an event by event basis, this method should give a correct result.

4 Trying the Average Weight Method on the MET Trigger

As a check of the average weight method, we calculated the differential Missing Energy rate (rather than integrating to get a trigger rate as a function of threshold). As stated before, the calculation correctly reproduces the (assumed) 32 MHz crossing rate. The fear is that the calculation with average weights will over estimate the rate a larger missing energy, where multiple events contribute to the trigger. The figure shown below compares the weighted result (black) to pure pileup (red) and to the “first event method” (blue).



The overestimation is visible but not too large and, at high values of the Missing Energy, the weighted calculation gives much smoother distribution. The pileup events reach the one event level then disappear. The "one event method" also suffers a big drop when the pileup event disappear because the binned events are basically only used to give $\frac{1}{17}$ of the large missing energy signal. For the weighted method also sees a drop as the pileup events disappear but it is not so severe. The pileup events do have too high a weight, indicating we need more of them, however, I expect the situation to improve when event by event weighting is possible. the pileup events giving large missing energy should get smaller weights.

5 How Well are We Sampling?

A small modification of the average weight program allows us to calculate the expected number of crossings of a specified type in the HLT sample. We can use this to see how well certain event types are represented in the sample. I did a small study of this as shown in the table below.

Events in Bins (0-11)	Rate (Hz)	Equiv. Pileup (Million Crossings)
$n_5 = 1$ $n_6 = 1$	400	21
$n_6 = 1$ $n_7 = 1$	100	83
$n_8 = 1$	19000	37
$n_3 = 2$ $n_4 = 2$	11000	6.2
$n_4 = 2$ $n_5 = 2$	17	13
$n_3 = 3$ $n_4 = 2$	850	7.4
$n_6 = 2$	300	32

Table 2: Rate in Hz and Equivalent pileup sample for some specific crossing requirements. We require exactly n_i events in bin i for one or two bins and sum over numbers in the other bins.

I tried to pick crossing configurations that had a reasonable rate, from a few Hz to a few hundred Hz. In every case tested, the binned HLT sample was equivalent to several million pileup crossings. This indicates the statistical power of the binning method. Even crossing patterns with multiple high $\hat{P}_T$ events are very significantly enhanced in the HLT sample compared to pure pileup. The lowest equivalent crossing numbers come from configurations with multiple events in the lower bins which are relatively plentiful even in the pileup sample.

My conclusion here is that the HLT crossings, binned as they are, are statistically quite powerful. If we use this statistical power correctly, our sample is equivalent to about 0.5 seconds of pure pileup crossings.

6 Conclusion

The weights calculated here will allow us to make use of the HLT production to determine trigger rates with good accuracy, even those rates dominated by multiple events per crossing adding to produce a trigger. The basic weight calculation involves no additional approximation, but, requires information about the produced events.

Using average weights per bin of produced events, again gives statistical power but may overestimate rates in the multiple event dominated triggers. We have looked at the most difficult trigger, Missing Energy, and found that the approximation only slightly over predicts rates, but, gives good statistical errors.

The weights given should give systematically better results for other triggers, which are dominated by single events in a crossing.

If we use the weighting method, the binned sample we have is approximately equivalent to 0.5 seconds (16 million) of pure pileup events.